Name (As It Appears on Gradescope):

Preferred Name:

ID:

CSCI 3104, Algorithms **Profs. Chen & Grochow**

Quiz 3E **Spring 2020, CU-Boulder**

Instructions: This quiz is a closed book and individual effort. Electronic devices (including, but not limited to phones, smart watches, FitBits, calculators, etc.) are NOT allowed. Possession of such electronics is grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

Please provide these:
Left neighbor name :
Right neighbor name :

Name (As It Appears on Gradescope):

Preferred Name:

ID:

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Quiz 3E **Spring 2020, CU-Boulder**

1. **Standard 5.** Write down a recurrence relation describing the number of steps the following algorithm performs on an input of size n.

```
// A is an array of size n
f(A[1, ..., n]):
    if len(A) > 1:
        for(i=1; i < floor(len(A)/2); i++):
            swap A[i], A[len(A)+1-i]
        f(A[1, ... , floor(len(A)/5)])
        f(A[ceil(len(A)/5), ..., n])
    else:
        return A
```

Solution: We note the non-recursive work consists of the `for` loop, which makes $n/2$ iterations. Each iteration takes a constant amount of work. So the non-recursive work is $\Theta(n)$.

Now there are two recursive calls: the first call has an array of size $n/5$, while the second call has an array of size $4n/5$. So our recurrence is:

$$T(n) = \begin{cases} T(n/5) + T(4n/5) + \Theta(n) & : n > 1, \\ 1 & : \text{otherwise.} \end{cases}$$

Name (As It Appears on Gradescope):

Preferred Name:

ID:

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Quiz 3E **Spring 2020, CU-Boulder**

2. **Standard 6.** Using the plug-in/unroll/substitution method, solve the following recurrence relation.

$$T(n) = \begin{cases} 2 & : n < 5, \\ 2T(n-5) + 3 & : n \geq 5. \end{cases} \tag{1}$$

Solution: Note that evaluating the recurrence yields the following:

$$\begin{aligned} T(n) &= 2T(n-5) + (1)3 \\ &= 2\Big(2T(n-10) + 3\Big) + 3 = 2^2T(n-10) + (1+2)3. \end{aligned}$$

Note in the $(1+2)3$ term, the 1 comes from the $(1)3$ term in the first line, and the coefficient of 2 comes from distributing the 2 in the $2\Big(2T(n-10)+3\Big)$ term.

Remark: We explicitly did not add the coefficients in the $(1+2)3$ term, as this will help us to spot a pattern.

Now we have that:

$$\begin{aligned} 2^2T(n-10) + (1+2)3 &= 2^2\Big(2T(n-15) + 3\Big) + (1+2)3 \\ &= 2^3T(n-15) + (1+2+2^2)3. \end{aligned}$$

Proceeding in this manner, we have that:

$$T(n) = 2^kT(n-5k) + 3\sum_{i=0}^{k-1} 2^i.$$

Now we determine the value of k that will get us to the base case. We note that we hit a base case precisely when $n - 5k < 5$. So:

$$\begin{aligned} n - 5k < 5 &\iff \\ n - 5 < 5k &\iff \\ k > \frac{n-5}{5}. \end{aligned}$$